

SISTEMAS DISTRIBUIDOS APLICADOS À AUTOMAÇÃO INDUSTRIAL

Autor: José Renato Gonçalves dos Santos
FDTE

Fundação para o Desenvolvimento Tecnológico da Engenharia
Caixa Postal 11.455 - SP - Brasil.

RESUMO: As necessidades de flexibilidade, disponibilidade, desempenho, manutenibilidade, modularidade e baixo custo, são fatores que têm favorecido a implementação dos sistemas modernos de automação industrial por sistemas distribuídos. Neste trabalho pretende-se dar uma visão geral das várias funções e dos vários componentes destes sistemas, ressaltando seus aspectos principais. Além disto, é descrito rapidamente no fim do trabalho, a implementação de um núcleo de sistema operacional distribuído, desenvolvido pela FDTE.

1. INTRODUÇÃO

Atualmente, muitos esforços têm sido desenvolvidos na área automação industrial.

A automação tem como objetivo, aliviar o operador humano de tarefas repetitivas e as vezes perigosas, deixando que estas tarefas sejam executadas automaticamente, por sistemas de processamento de dados. Além disto, a automação permite a realização de funções que não eram realizadas antes, tais como, manipulação de dados históricos dos processos, geração automática de relatórios, cálculos de estatísticas do processo, etc. A aplicação da automação na indústria, permite entre outras coisas, o aumento da produção, a minimização dos custos, o aumento da eficiência do processo, o aumento da segurança e confiabilidade, etc.

Dentro deste panorama, a utilização de sistemas distribuídos, para controle e automação industrial, tem sido uma solução adotada muito frequentemente, nos dias atuais.

2. SISTEMAS DISTRIBUIDOS

O tema "distribuído", é um assunto polêmico e é encontrado na literatura sob diferentes enfoques. Diferentes autores, abordam o conceito de "sistema distribuído" com significado e abrangência diferentes. Assim, um determinado sistema pode ser considerado "distribuído", pela definição de um determinado autor e ao mesmo tempo não ser "distribuído", segundo um outro autor. É importante, então, definir o conceito de "sistema distribuído", dentro do contexto deste trabalho.

Uma máquina de arquitetura distribuída, pode ser caracterizada como um conjunto finito de módulos de processamento/armazenamento (nós) com autonomia para execução concorrente e com capacidade para troca de informações [5][6].

Um sistema de processamento distribuído, deve necessariamente, ser suportado por uma máquina de arquitetura distribuída. Porém somente isto, não é suficiente, devendo o sistema possuir algumas características adicionais, para poder ser considerado "distribuído".

A primeira característica importante de um sistema distribuído, é que nenhum dos nós da máquina de arquitetura distribuída deve concentrar o controle da execução das funções do sistema, mas sim, estas funções devem ser executadas de forma distribuída, dentro dos diversos nós da máquina. Esta característica é fundamental, para a confiabilidade do sistema. Desta forma, a falha de um determinado nó, não deve causar a parada total do sistema, mas sim causar apenas uma degradação dos serviços e funções executadas.

Outra característica dos "sistemas distribuídos" é a substituição do conceito de estado global pelo conceito de estado local de um nó. Assim como o controle, as informações do estado da máquina também devem ser distribuídas pelos vários nós. Desta forma os vários módulos do processamento/armazenamento só têm acesso aos seus estados locais e podem estimar os estados locais dos demais módulos através da troca de informações entre nós. A estimativa do estado global da máquina é um parâmetro aproximado, devido as características físicas do sistema de comunicação entre nós (atrasos, erros de transmissão, etc)

Os sistemas distribuídos devem também obedecer uma política de autonomia cooperativa. Os diversos nós da máquina, apesar de autônomos e independentes, devem cooperar entre si, com a finalidade de realizar uma determinada função global do sistema. Deve existir uma política global, a ser obedecida pelos vários nós, para que as funções do sistema possam ser realizadas de forma adequada.

Podemos então, identificar um "sistema distribuído" através de suas características básicas:

- multiplicidade de módulos de processamento/armazenamento
- sistema de interconexão entre módulos
- controle distribuído
- conceito de estado local
- autonomia cooperativa

3. AUTOMAÇÃO INDUSTRIAL

O emprego de computadores para aplicações de controle e automação industrial, é uma realidade dos dias atuais.

Os processos produtivos industriais são processos naturalmente distribuídos e a utilização de "sistemas distribuídos de controle digital" (SDCD) é uma solução natural. Nestes sistemas o processamento se dá de forma paralela entre os diversos nós do sistema distribuído, havendo pontos de sincronização, nos instantes onde há troca de informação.

Apesar do SDCD ser um sistema tipicamente "distribuído", as várias funções da automação e controle podem ser agrupadas em vários níveis hierárquicos, conforme a fig. 3.1. [1]

O nível de processo é responsável pelo controle propriamente dito, aquisitando dados e acionando elementos da planta, aplicando algoritmos específicos de controle, definindo valores de variáveis de processo, etc.

O nível de coordenação é o responsável pelo gerenciamento de recursos do processo, para o primeiro nível, além de permitir a monitoração das variáveis e o estabelecimento dos pontos de operação do processo.

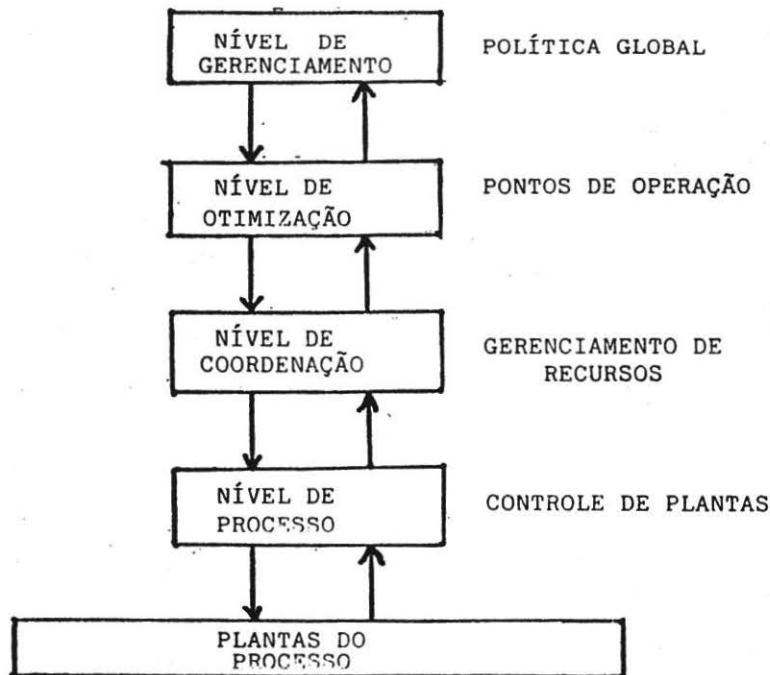


Fig.3.1 - Níveis Hierárquicos de Controle

No nível de otimização, são gerados os pontos de operação e demais parâmetros necessários a obtenção do melhor rendimento possível, através da aplicação de valores reais de variáveis nos modelos matemáticos dos processos.

O nível de gerenciamento é responsável pela determinação da política global para a operação produtiva, levando em conta os recursos físicos instalados e disponíveis, as matérias primas e energia, os recursos humanos, a política da empresa no setor e a legislação que regulamenta a produção e o produto.

Os diversos níveis executam desta forma, funções diferentes e obedecem a uma estrutura hierárquico onde os níveis superiores definem parâmetros para os níveis inferiores. Apesar desta classificação hierárquica, o mal funcionamento de algum dos níveis, não deve afetar o funcionamento dos níveis inferiores, que estão mais próximos do processos físico. Neste caso os parâmetros definidos pelos níveis superiores devem poder ser definidos pelo operador ou o sistema deve adotar algum valor provisório para eles até

a normalização do funcionamento.

4. COMPONENTES DO SISTEMA DISTRIBUÍDO

As funções implementadas por um Sistema Distribuído podem ser divididas em vários níveis hierárquicos conforme a fig. 4.1.



Fig.4.1
Níveis do Sistema
Distribuído

Cada nível hierárquico executa uma série de funções utilizando funções dos níveis inferiores. Desta forma, cada nível hierárquico implementa uma máquina virtual (MVi), que executa as funções das máquinas virtuais implementadas pelos níveis inferiores acrescidas pelas funções implementadas por ele próprio. [6]

4.1. MÁQUINA DE ARQUITETURA DISTRIBUÍDA

O hardware constituído pela máquina de arquitetura distribuída implementa o primeiro nível de um sistema distribuído.

Vários tipos de equipamentos podem ser utilizados como máquinas de arquitetura distribuída. Uma solução, adotada largamente na implementação de sistemas distribuídos, principalmente em aplicações de automação, tem sido o uso de redes locais de micro-computadores. As funções de controle, coordenação, otimização e gerenciamento, assim como as demais funções da fig.4.1, podem ser implementadas por diversos micro-computadores, que serão distribuídos pela fábrica e interligados por uma rede local.

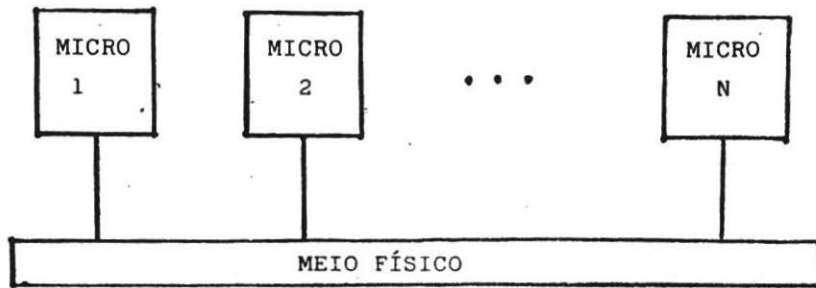


Fig.4.2 - Arquitetura de Rede Local

A existência de vários microcomputadores e redes locais no mercado nacional, assim como a grande quantidade de software disponível para estes sistemas, justificam a sua utilização como infra-estrutura para sistemas distribuídos.

Outra grande vantagem do uso de microcomputadores é que estes sistemas, já possuem acoplados elementos periféricos como monitores de vídeo, teclado, impressora, drives de disco, etc; com uma interface de acesso disponível através do sistema operacional local.

4.2. SISTEMA OPERACIONAL DISTRIBUÍDO

Um sistema operacional convencional, é o software básico, responsável pela implementação de funções e serviços que simplificam a interação dos usuários com o sistema. A principal função do sistema operacional é gerenciar os diversos recursos oferecidos pelo sistema, e oferecer uma interface simples e poderosa, que permita o acesso a estes recursos pelo usuário. Estes recursos englobam tanto os recursos físicos oferecidos pela máquina (processador, memória primária, memória secundária, periféricos, etc) como também recursos lógicos criados pelo próprio sistema operacional (arquivos, base de dados, etc...)

Entendemos por sistema operacional distribuído, um sistema operacional desenvolvido para uma máquina de arquitetura distribuída. A sua função básica é gerenciar os vários recursos do sistema que se encontram distribuídos pelos vários módulos de processa-

mento/armazenamento da máquina; e também oferecer ao usuário, uma interface global unificada, do sistema.

Em certos casos, é desejável que o usuário não precise se preocupar com a localização física dos recursos da máquina, tendo acesso a estes, através de uma interface transparente. Em outros casos, porém, o usuário pode desejar especificar a localização de um determinado recurso para que possa otimizar a operação do sistema, ou por outro motivo qualquer. O ideal é que o sistema operacional ofereça os dois tipos de serviços (transparente ou não) em sua interface com o usuário.

Dentro disto, podemos considerar o "sistema operacional distribuído" como o elemento responsável pela integração do sistema, transformando uma coleção de módulos autônomos e independentes de processamento em um sistema unificado do ponto de vista do usuário.

Em aplicações de automação industrial, além da característica de sistema distribuído, o sistema operacional deve apresentar também características de sistemas de tempo real. Neste tipo de sistema, uma das grandes preocupações é minimizar os atrasos das atuações e monitorações de dados, em relação aos instantes ideais que tais operações devem ser realizadas. [2]

O sistema operacional distribuído pode ser dividido em dois sub-níveis: núcleo e nível de sistema conforme a figura 4.3.

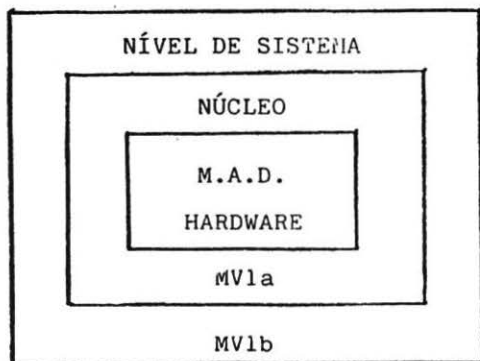


Fig.4.3 - Sub-níveis do Sistema Operacional Distribuído

4.2.1. NÚCLEO DO SISTEMA OPERACIONAL DISTRIBUÍDO

O núcleo do sistema operacional distribuído é o responsável pela criação de um ambiente de multiprogramação e multiprocessamento. Ele é o responsável pela manipulação de processos concorrentes.

Uma das funções do núcleo, é gerenciar o uso do processador, de um nó de processamento, pelos vários processos residentes naquele nó. Este gerenciamento deve ser baseado em uma política de prioridades que permita que processos que devam tratar de aspectos de tempo real, sejam altamente prioritários em relação a outros processos.

Outra função implementada pelo núcleo é o sincronismo de processos residentes em um mesmo nó do sistema.

Em sistemas de tempo real, o núcleo deve implementar também, funções de relógio e temporização de eventos.

Uma outra função do núcleo, muito importante, é o mecanismo de troca de mensagens. Este mecanismo é a única forma de processos residentes em nós distintos poderem trocar informação e se sincronizarem. Isto é essencial para que os níveis superiores do sistema distribuído possam implementar as funções do sistema de forma distribuída.

4.2.2. NÍVEL DE SISTEMA

O nível de sistema é o responsável pela implementação das funções de gerenciamento dos recursos do sistema. Basicamente, identificamos no sistema quatro gerenciadores de recursos diferentes: gerenciador de memória, gerenciador de processadores, gerenciador de dispositivos e gerenciador de informações.

Como todas as outras funções do sistema, as funções de gerenciamento, também devem ser implementadas de forma distribuída. Isto pode ser feito através de vários processos servidores, residentes nos vários nós do sistema, comunicando entre si e com os processos do usuário, através do sistema de comunicação implementado pelo núcleo. Através deste mecanismo os vários gerenciadores do sistema, devem implementar políticas de controle distribuído, o que envolve funções dinâmicas, democráticas, descentralizadas, mu

tiplicadas, adaptativas e estocásticas. Esta forma de controle não é totalmente formalizada, devendo o projetista deste sistema utilizar principalmente a sua intuição e experiência de projeto.

A interface que o nível de sistema oferece aos níveis superiores, deve permitir o acesso aos vários recursos do sistema de forma que o usuário possua uma visão unificada e transparente do sistema, não precisando se preocupar com a localização física dos recursos. Porém, a possibilidade de especificação da localização do recursos também deve existir, para casos em que o usuário julgar isto necessário.

4.3. NÍVEL DE APOIO

Neste nível são implementadas funções complementares, às funções do sistema operacional, como a criação de ambiente para desenvolvimento de programas, implementação de ferramentas de auxílio a depuração e testes de programas, implementação de rotinas de biblioteca, etc.

Em aplicações de automação e controle, os programas aplicativos, são em geral, desenvolvidos em outros sistemas. Estes sistemas de desenvolvimento, devem oferecer uma série de recursos, tais como tradutores de linguagem (compiladores, tradutores, montadores), construtores de programas executáveis (ligadores, relocadores, carregadores) e programas de apoio ao desenvolvimento (editores, simuladores).

Estas funções podem também ser implementadas no próprio sistema de automação, durante a sua operação em tempo real. Isto é feito através de processos de baixa prioridade de forma a não atrapalhar o funcionamento dos processos de tempo real. A este tipo de operação dá-se o nome de "apoio ao desenvolvimento" ("back ground"). [2]

É ainda extremamente conveniente, que o sistema de desenvolvimento possua ferramentas para desenvolvimento de programas distribuídos, já que o programa aplicativo de controle e automação é basicamente um programa distribuído.

Ferramentas de auxílio a depuração e teste de programas são essenciais para implantação de um sistema distribuído de automação. A tarefa de depuração e teste de programas convencionais, já é

uma tarefa difícil, e em geral consome grande parte do tempo de desenvolvimento. Em ambientes distribuídos esta tarefa é ainda mais difícil, já que um programa distribuído é muito mais complexo e apresenta vários aspectos que dificultam a sua depuração.

As rotinas de biblioteca também são muito úteis para os programas aplicativos, que podem utilizar rotinas prontas para efetuarem tarefas mais comuns. Dentre alguns tipos de rotinas podemos citar: algoritmos de controle (PID, PI, etc), conversão de formatos de dados (ASCII, decimal, BCD, etc), manipulação de string, operações em ponto flutuante, operações trigonométricas, etc.

4.4. NÍVEL DE APLICAÇÃO

Neste nível são executadas as funções específicas de controle e automação do sistema. Este nível é o responsável, pela implementação das funções dos níveis de processo, coordenação, otimização e gerenciamento. Estas funções encontram-se distribuídas pelos nós do sistema e cada módulo pode implementar uma ou várias destas funções.

É função do nível de aplicação, efetuar o controle e a aquisição de dados das plantas dos processos, otimizar a operação do processo, gerenciar a base de dados distribuída, que armazena as informações do processo, emitir relatórios, e também criar uma interface de processo com o operador, eficiente.

As funções específicas de controle, devem ser executadas em tempo real por processos altamente prioritários, que devem ser executados com o mínimo atraso possível.

A interface com o operador deve permitir a monitoração das variáveis do processos, oferecendo uma visão global e eficiente do processo. Além disto a interface deve permitir a atuação do operador sobre o processo através de vários comandos específicos. Esta interface, na realidade, é composta por um conjunto de interfaces locais a cada nó. Assim, as interfaces locais permitem a monitoração e comando de aspectos locais do processo de acordo com o nível hierárquico específico implementado por aquele nó (processo, coordenação, otimização, gerência).

De um bom projeto da interface homem/máquina, irá depender o desempenho de todo o sistema. O operador deve ter condições de in

teragir com o processo de uma maneira simples e de fácil aprendizado.

5. SC - UM NÚCLEO DE SISTEMA OPERACIONAL DISTRIBUÍDO

O SC (Sistema de Comunicação) é um núcleo de sistema operacional distribuído, desenvolvido pela FDTE, para aplicações de tempo-real. [16]

Existem atualmente duas versões do SC: SC85 e SC86. O SC85 é a versão do SC para máquinas baseadas nos microprocessadores 8080 e 8085 da Intel e Z80 da Zilog. Já o SC86, é a versão para máquinas baseadas nos microprocessadores 8088 e 8086 da Intel.

A principal função do SC é criar um ambiente de multiprogramação e multiprocessamento. Ele possibilita a execução concorrente de vários processos residentes em um mesmo nó ou em nós distintos.

A CPU de um determinado nó é compartilhada pelos processos residentes naquele nó, baseado nas suas prioridades. Os processos ativos, podem estar em três estados possíveis: bloqueado, pronto, ou executando. No estado bloqueado, o processo se encontra na fila de um recurso (semáforo) a espera da sua liberação. No estado de pronto o processo encontra-se na fila da CPU, esperando sua vez para ser executado. O primeiro processo desta fila, é o processo que se encontra no estado de execução daquele nó. Tanto a fila da CPU, como as filas dos semáforos, são organizadas de acordo com as prioridades dos processos.

Além disto, o SC executa uma série de funções que são implementadas através de chamadas de primitivas (rotinas) do SC. Estas funções incluem:

- a) ativação e desativação de processos
- b) sincronismo e controle de recursos locais (semáforos)
- c) temporização de eventos
- d) troca de mensagens

O mecanismo de ativação permite que um processo que já esteja residente na memória seja ativado, no SC.

O sincronismo e controle de recursos locais de um nó, são feitos pelo mecanismo tradicional de semáforos e primitivas P e V (pedido e liberação de recurso).

A temporização de eventos permite que o usuário programe um temporizador, gerenciado pelo SC com um determinado valor, e indique o endereço de uma rotina que deve ser chamada caso estoure a temporização.

A troca de mensagens é feita entre processos residentes em um mesmo nó ou em nós distintos de forma idêntica. Cada processo é identificado por um número de processo local e um número de processador, sendo desta forma univocamente identificado no sistema.

As várias primitivas do SC são:

- a) primitivas de ativação e desativação de processo
 - ATVUS: ativa processo de usuário
 - EXIT : desativa processo corrente
- b) primitivas de sincronização e controle de recursos
 - PED\$SMF: pede semáforo
 - LIB\$SMF: libera semáforo
 - P : pede recurso
 - V : libera recurso
- c) primitivas de temporização
 - PED\$TMP: pede temporizador
 - LIB\$TMP: libera temporizador
 - TEMPO : liga temporizador
 - DESLIG : desliga temporizador
- d) primitivas de troca de mensagens
 - PED\$BP: pede buffer para montagem da mensagem
 - LIB\$BP: libera buffer
 - ENVIA : envia mensagem
 - SPERA : espera mensagem
 - EXIST : verifica se existe mensagem para o processo corrente.

Além do SC, foi desenvolvido também, um monitor do SC cujo principal objetivo é auxiliar na depuração e testes de programa distribuídos implementados sobre o SC.

O monitor permite que seja feito um "trace" dos eventos do SC ocorridos durante o processamento, através de um monitor de vídeo. Estes eventos do SC, compreendem: a escalação dos processos e todas as chamadas de primitivas efetuadas por processos em execução.

O monitor permite ainda, que seja feita uma execução passo a passo, onde após cada evento, pode-se executar comandos implemen-

tados pelo monitor. Entre estes comandos, incluem-se o envio de mensagens através do teclado, à visualização dos estados das estruturas internas de dados do SC, tais como, filas, semáforos, temporizadores, etc.

O monitor é uma ferramenta muito útil, para o teste de programas distribuídos, pois permite a visualização do programa, como uma sequência de eventos, em cada nó do sistema.

6. CONCLUSÃO

As várias funções de um sistema distribuído podem ser divididas em vários níveis: hardware, sistema operacional, nível de apoio e nível de aplicação. O sistema operacional por sua vez, pode ser subdividido em núcleo e nível de sistema.

Esta estruturação permite a separação das funções de infraestrutura do sistema, que são implementadas pelos níveis inferiores, das funções específicas da aplicação. Desta forma o nível de aplicação só precisa se preocupar com as tarefas de automação e controle, deixando as tarefas básicas do sistema, serem implementadas pelos níveis inferiores.

7. BIBLIOGRAFIA

- [1] Carvalho, Antonio J.C. & Aguilera, Edson F., "Aplicações de Sistemas Distribuídos de Controle Digital - SDCD", Anais do 6º Congresso Brasileiro de Automática, 1986
- [2] Neto, João J., "Aspectos do software básico para Sistema de Controle de Processos por Computador", Anais do 1º SICOP, 1981
- [3] Melnikoff, Selma S.S., "Aspectos de Programação em Software de Aplicações de Controle de Processos", Anais do 1º SICOP, 1981
- [4] Garcia, José Dalvio G., "Pacotes de Aquisição de Dados e Controle direto", Anais do 1º SICOP, 1981
- [5] Ruggiero, Wilson V., "Arquitetura de Sistemas Distribuídos para Controle de Processos", Anais do 1º SICOP, 1981
- [6] Bressan, Graça, "Ambiente de Programação Distribuída: Definição, Análise e Síntese", Tese de doutoramento apresentada à Escola Politécnica da USP, 1985.
- [7] Enslow, Philip H., "What is a Distributed Data Processing Sys-

- tem" Computer, jan. 1978
- [8] Forsdick, Harry C.L.Schantz, R.E. & Thomas, R.H., " Operating Systems for Computer Networks", Computer, jan. 1978
- [9] Stankovik, John A., "A perspective on Distributed Computer Systems", IEEE Transactions on Computers, vol.33, nº 12, dez - 1984
- [10] Donneley, Jed, "Componentes of a Network Operating System", Computer Networks, vol.3, nº 6, dez.1979
- [11] Bacon, Jeam, "An Approach to Distributed Software Systems", ACM Operating Systems Review, vol.15, nº 4, out.1981
- [12] Rashid. R.F. & Robertson, G.G., "Accent: A communication oriented network system kernel", ACM Operating Systems Review, Vol.15, nº 5, dez.1981
- [13] Craft, Daniel H., "Resource Management in a Decentralized System", ACM Operating Systems Review, vol.17, nº 5, out.1983
- [14] Martucci, Moacyr Jr., "Especificação da Arquitetura de um Sistema Distribuído para controle de processos" - Tese de doutoramento apresentada à Escola Politécnica da USP, 1982.
- [15] Massola, A.M.A. & Cugnasca, C.E. & Chagas, A.J. & Fumis, H.P. "Sistema de automação modernos: uma aplicação na indústria de alumínio - Revista Politécnica nº 194
- [16] FDTE; "Manual do usuário do SC", publicação interna